# INTEGRATED DATA TRAFFIC MONITORING SYSTEM

**Inventors:**

**Robert J. Demopoulos**

**David J. Fladebo**

**Assignee:**

**The Barrier Group**

Merchant & Gould P.C.

P.O. Box 2903

Minneapolis, MN 55402-0903

# INTEGRATED DATA TRAFFIC MONITORING SYSTEM

## Related Applications

This application claims priority of United States provisional application Serial Number _____, entitled INTEGRATED DATA TRAFFIC MONITORING SYSTEM, filed January 23, 2004 (also identified by attorney docket no. 14584.0004USP1), which is hereby incorporated by reference.

## Technical Field

This application relates generally to monitoring data traffic related to computing systems, and more particularly to an integrated system for monitoring data traffic.

## Background

Security is now a very important aspect of any computing system connected to the Internet. In order to provide protection from different types of security threats, a typical computing system may employ a significant number of technologies to monitor the computing system and, in some cases, perform actions to protect the computing system from identified threats or potential threats. These technologies will be referred generally throughout this specification as monitor modules. Some common monitor modules and their functions include:

- Stateful firewall – An industry standard method of network connection monitoring, control and protection
- Application awareness – Inspecting network connections for proper application behavior protecting a network from common application vulnerabilities
- DHCP – Provides IP address and other network parameters to network users
- IDS – Intrusion Detection System, detects attacks
- IDP – Intrusion Detection and Prevention, detects and prevents attacks
- HIDS – Host-based Intrusion Detection systems, detects attacks and changes on the security device itself

- Service proxy and cache server – Isolates users from the Internet, controls their access and improves speed of Internet use
- Email forwarder with masking – Isolates and controls incoming or outbound Email
- WEB forwarder with masking – Isolates, protects and controls incoming or outbound WEB service requests
- Anti-SPAM - Prevents the majority of unsolicited Email requests
- Web content filter – Protects organizations from access to or from unacceptable WEB sites and content
- Anti-virus filter – Examines incoming Email and other services for the presence of viruses and removes them
- Email content filter – Controls the content of Email messages to protect against SPAM and unacceptable content
- Multiple DMZ – The ability to segregate a customer's network into isolated "De-Militarized Zones", provides protection by isolation
- VPN Concentrator – Allows for connection from anywhere in the world to a "Virtual Private Network" that from a remote site appears as a single network segment
- VPN Initiator – Connects to other VPN concentrators
- Site-to-site VPN with full mesh option – Allows for the creation of large private network utilizing inexpensive public Internet connections. Useful for companies with small branch or remote offices/locations
- Encryption at all levels – All data transferred or stored in an encrypted or encoded format
- Honey Pot – A method to trap intruders and to track attackers
- SSH/SSHD - A secure method of communicating and managing security appliances and services
- Automatic updates via WEB - Self-maintaining, correcting, updating and reporting mechanisms
- HA/Cluster implementation – High-availability redundant capability that can grow as required depending on performance requirements

- Common web-enabled management interface – All technologies and services are managed by a common WEB based interface
- SAMBA, LDAP support – Windows network file system and user awareness
- Full identification, authentication and authorization (AAA) support – Method to ensure proper user access and logging of user connection to network resources
- Multi factor identification required for device management – More extensive methods used for administrative access to security devices for management and control.
- SNMP device inspection and control – The ability to query and control devices such as routers, switches, printers, workstations and printers to gather detailed network information without the need for a device specific resident client.
- Clear text password detection – The ability to detect, log and report the use of internal or external usernames and passwords that are not encrypted (clear text).

Monitor modules such as those described above each perform a different monitoring and/or security function and are usually provided as a separate and distinct application (or device, depending on the implementation) on the computing system. Because computing system administrators wish to select and employ only those monitor modules deemed necessary, most monitor modules are designed to be standalone modules that function independently of the existence of other monitor modules. Therefore, each monitor module independently generates and tracks various data as necessary to perform its function, regardless of whether the same data is being tracked or generated by other monitor modules.

In addition, because the developer of a monitor module cannot rely on the existence of other monitor modules or even a common data format for data generated by other data systems, most monitor modules are not designed to interface with other monitor modules or even provide data in A format useful to other monitor modules. Therefore, monitor modules are not capable of taking advantage of information known to other monitor modules or reacting to actions being performed by other monitor modules.

For example, an anti-virus filter might include a file of known viruses that it uses when screening message traffic received by the computing system. Any messages containing files that include a virus identified in the known virus file is deleted, quarantined, or otherwise acted on by the

virus filter without input from, or knowledge of, the other monitor modules. Similarly, an anti-spam filter may include a list of words or other information that it uses to screen out messages received by an e-mail application. These monitor modules may report data to an administrator of the computing system indicating that viruses or spam have been detected or that actions have been taken, but the other monitor modules on the computing system are unaware of and make their own decisions independent of any such knowledge or actions. It is left to the administrator to determine from the data if another monitor module needs to be provided with this new data to more effectively perform its function.

Each disparate monitor module has its own requirements for evaluating messages received from the communication network. In the case of an anti-virus filter, the entire message is typically received before the filter makes its analysis. The same is true for the anti-spam filter. A firewall, on the other hand, can delete the packets that make up a communication as the packets arrive, preventing them from ever being passed into the computing system proper. However, the firewall has no way of predicting that a given message or communication contains a virus, is spam, is an attempt to take over the computer, or represents some other threat, so such threats are passed into the computer to be screened by the other monitor modules.

Because the monitor modules do not share information, the fact that threats are identified by one monitor module, does not benefit any of the other monitor modules. Take, for example, a situation where a remote computer is attempting to take control of a computing system. The first effort may be to infect the computing system with one of a number of viruses that allow remote control of the computing system, by sending virus-laden messages to the computing system. If the virus software catches all of the viruses, then an attempt may be made to log into the computing system as a user. If the clear text password detection system foils this attempt, an attempt may then be made to reconfigure the computing system to allow public access to restricted material, thereby testing the HIDS system. This scenario shows that if the remote computer keeps looking for weaknesses long enough, it is likely something will be found. As the monitor modules do not interface with each other, the password detection system does not have the benefit of the knowledge that there have already been repeated infection attempts from the remote computer. Similarly, the HIDS system does not know that the remote computer was the source numerous, different, and concerted attempts to take over control of the computer.

The monitor modules often report data related to identified threats and the actions taken in response to an administrator. However, it is up to the administrator to read the disparate reports and notifications and attempt to identify trends indicative of a more significant threat to the computing system. In the scenario described above it is left to the administrator to view the data from each of the monitor modules, correlate the data, determine an appropriate coordinated response by the computing system, and implement the response. Depending on the level of communications traffic and size of the computing systems, this may involve the analysis of huge amounts of data stored in multiple data logs, each in different formats and containing different types of information. The administrator may have difficulties correlating data from one monitor module to data from another monitor module, not to mention difficulties in identifying trends in the collected data.

The scenario described above used a relatively simple example where all the attacks are coming from one remote computer. Other scenarios are possible where the attacks have other, but less obvious, common characteristics such as they all have the same destination, subject line or some other attribute. Such information may not even be tracked by each monitor module and may only be determinable upon review of a collected and correlated set of data from all the monitor modules.

Administrators have a further challenge in that most attacks occur quickly. Often, by the time the administrator has determined from the data provided by the various monitor modules that a concerted attack on multiple fronts is occurring, it has either succeeded or failed. Administrators cannot analyze the data provided in time necessary to provide effective feedback to the various monitor modules.

In reality, even though a plethora of threat data exists and is being reported in real time, it is typically used after the fact to determine what occurred after a successful attack.

## Summary

The present invention includes an integrated data traffic monitoring system monitoring data traffic on a computer system. The integrated data traffic monitoring system includes a security appliance and one or more security and monitoring technologies such as high performance commodity hardware and open source and proprietary software products. The security appliance and the security and monitoring technologies may be implemented as separate and distinct modules or combined into a single security appliance. The security and monitoring technologies monitor

network data traffic on, or directed to, the computer system. The integrated data traffic monitoring system collects data from each of the security and monitoring technologies into an event database. The contents of the event database are then continuously analyzed for possible threats to the computer system. In some cases, the analysis may determine that certain messages or messages with certain attributes are potential threats and feedback, in the form of alerts, notifications, commands, or instructions, may be provided to one or more of the security technologies.

The integrated data traffic monitoring system may be implemented on a single computing system. The integrated data traffic monitoring system also may be distributed so that data traffic on multiple computing systems is monitored. In the distributed system, data from security and monitoring technologies transmit data to a security appliance on a remote computing system. The security appliance collects the data and analyzes it at the remote computing system and potentially may provide feedback to the computing systems being monitored.

In accordance with other aspects, the present invention relates to a method of identifying threats to a computing system received from a network. The computing system includes a plurality of monitor modules including a first monitor module that evaluate a message received from the network. For a message identified as a potential threat, the system receives event data related to the message from the one or more monitor modules. The event data is stored in a first event record in a database on the computing system that already includes a plurality of second event records of event data related to previous messages. After receipt of event data from any one of the plurality of monitor modules, the first and second event records in the database are analyzed. Based on the results of the analysis, a command is transmitted to the first monitor module, in response to results of the analysis, the command including at least some event data from the first event record and a security action to be taken by the first monitor module.

In accordance with yet other aspects, the present invention relates to a method of monitoring communication traffic on a network. The method includes evaluating, by a first computing system and a second computing system, a first message received from the network. The first computing system generates new event data related to the first message including a site priority determined by the first computing system. The new event data is transmitted to a security facility that stores the new event data in an event database that includes pre-existing event data. Based on the event data in

the event database, a network threat level for the message is calculated and commands are issued based on the network threat level from the security facility to the second computing system.

In accordance with yet other aspects, the present invention relates to a method for deleting messages received by a computing system from a network. The method includes receiving a message in a buffer on the computing system in which the message directed to a destination on the computing system. The message is evaluated with a plurality of monitor modules. If the message is identified as a potential threat by one or more of the monitor modules, The output of the monitor modules related to the message is stored in a new event record in a database containing a plurality of previous event records. The output may include event data describing attributes of the message, a threat type, and an assigned priority. The event records in the database are analyzed. The message is selectively deleted from the buffer before delivery to the destination.

In accordance with yet other aspects, the present invention relates to a computer readable medium having stored thereon a database of event records for each message received within a period of time by a computing system. The messages were previously identified as a threat by one or more monitor modules on the computing system and each event record includes event data provided by the monitor modules that identified the message as a threat. The event data stored in the event record includes a priority level of the message assigned by one or more monitor modules, an event identifier, an event type, an event description, an event priority, a date and time associated with the message, data identifying the message's source, and data identifying the message's destination.

## Brief Description of the Drawings

FIG. 1 illustrates an integrated monitoring system in accordance with an embodiment of the present invention.

FIG. 2 illustrates some of the functional components of an embodiment of an integrated monitoring system for a computing system.

FIG. 3 illustrates another embodiment of a multi-system implementation of an integrated monitoring system.

FIG. 4 shows an embodiment of the logical operations of an monitor module in an integrated monitoring system on a computing system such as the one described with reference to FIG. 2.

FIG. 5 shows an embodiment of the logical operations of an SSI in an integrated monitoring system on a computing system such as the one described with reference to FIG. 2.

FIG. 6 shows an embodiment of the logical operations of an SSMI in a multi-system integrated monitoring system such as the one described with reference to FIG. 3.

## Detailed Description

Various embodiments of the present invention will be described in detail with reference to the drawings, wherein like reference numerals represent like parts and assemblies throughout the several views. Reference to various embodiments does not limit the scope of the invention, which is limited only by the scope of the claims attached hereto. Additionally, any examples set forth in this specification are not intended to be limiting and merely set forth some of the many possible embodiments for the claimed invention.

In one possible embodiment, a computing system may include a single computing device or multiple, connected computing devices. Computing devices are electronic devices that perform functions using a combination of hardware and/or software. Computing devices may include such hardware as a processor, computer readable storage media (including, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by the system), and one or more communication devices suitable for transmitting and receiving data over communication media. In addition, computing devices may also include software, firmware or a combination of the two stored on the computer readable media. Examples of computing devices include personal computers, handheld computing devices, mobile communication devices, cellular telephones, networked appliances, computer servers, and mainframes and any other programmable device that is exposed to and receives data traffic.

Communication media includes any medium capable of carrying data or information such as computer-readable instructions, data structures, and program modules, whether such data is embodied in a modulated data signal such as a carrier wave or other transport mechanism. By way of example, and not limitation, communication media includes wired media such as a wired network

or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media.

Computing devices may be implemented using different software operating systems and programming languages. Examples of operating systems include Microsoft Windows XP, Macintosh OS X, OS2, Unix- and Linux-based operating systems, and Microsoft Windows CE. Examples of programming languages suitable for developing software embodiments include C, C++, Java, Visual Basic, Perl, and markup languages such as XML, HTML, and XAML. Selection of operating systems and software languages is often more an issue of user and developer preferences or convenience.

Computing devices may be described in terms of the logical operations performed by the devices. The logical operations of the following various embodiments are implemented (1) as a sequence of computer implemented acts running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations making up the embodiments described herein are referred to variously as operations, structural devices, acts or modules. It will be recognized by one skilled in the art that these operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims attached hereto.

FIG. 1 illustrates an exemplary computing system 100 that implements an embodiment of an integrated monitoring system 120. The exemplary computing system 100 as shown includes an email server 102, a web server 104 and an intranet server 106. The servers are further connected to an internal communication network 108, such as an intranet. The internal communications network 108 connects the various computing devices and components internal to the computing system 100. In the embodiment shown, the internal network 108 is connected to the servers 102, 104, 106 and a plurality of additional computing devices 112. The computing system 100 is further connected to other remote computing systems 124, 126 via an external communications network 122. The external communications network 122 may be the Internet or may be some other wired or wireless communications network.

In the environment shown in FIG. 1, communications traffic in the form of data transmitted on the network 122 may pass between the computing system 100 and the remote computing systems. In addition, there also may be communication traffic passing between various elements within the computing system 100. The communications traffic on the network 122 and within the computing system 100 will be discussed as consisting of a plurality of separate and identifiable "messages". Examples of messages on the Internet include, for example, digital files, email messages, web pages, voice over internet protocol (VOIP) data streams, and streaming audiovisual data. Messages are transmitted in digital form as one or more packets of digital data.

The embodiment in FIG. 1 also includes an integrated monitoring system 130, which monitors communication data traffic. The integrated monitoring systems 130 may be implemented to monitor data traffic on the internal network 110, data traffic received from the external network 122, or both depending on the implementation. The integrated monitoring system 130 analyzes the communication traffic in order to identify messages that may pose a threat to the computing system and block or quarantine any such messages identified. Such threats include any unwanted or undesirable occurrence related to data traffic such as, for example, spam, viruses, denial of service attacks, unauthorized attempts to infiltrate the computing system, etc. While some threats may be actual threats of harm or damage to the system, others may simply be inconvenient, annoying or unwanted events and not pose any risk of damage to the computing system. The integrated montiring system 130 will be discussed in greater detail with reference to FIG. 2 below.

FIG. 1 shows the integrated monitoring system 130 connected to the internal network 110. However, it should be noted that the integrated monitoring system 130 may be connected to the internal and external networks in many different ways and still perform its security functions. For example, in one embodiment the integrated monitoring system 130 is implemented as a gateway between the external communication network 122 and the internal communication network 110. Messages destined for the computing system 100 are screened by the integrated monitoring system 130 before being passed on to the internal network 110. In an alternative embodiment, all messages carried on the internal network, regardless of whether they originate from a computing device 112, a server 102, 104, 106 or the external network 122, pass through the integrated monitoring system 130.

FIG. 2 illustrates the functional components of an embodiment of an integrated monitoring system 200 for a computing system. The integrated monitoring system 200 includes multiple

monitor modules 202, 204, 206, 208, 210, and a security system integrator (SSI) 212. The SSI receives data reported from the monitor modules and may issue commands to at least some of the monitor modules. Embodiments of the SSI 212 may include such components as an analysis module 216 that analyzes the contents of an event database 214, an alerting module 218 that transmits security alerts (such as to system administrators and users), a command and control module 220 that provides an interface between the SSI 212 and the monitor modules 202, 204, 206, 208, 210, a communication module 224 that supports the reporting of the contents of the event database 214 to other locations, and a log database 222 that stores a log record of actions taken by the integrated monitoring system 200 over time. Each of these components is discussed in greater detail below.

The integrated monitoring system 200 includes a plurality of monitor modules 202, 204, 206, 208, 210. Each monitor module 202 may independently perform one or more different monitoring and security functions. The functions of some monitor modules also may overlap. In general, the monitor modules monitor and evaluate communications traffic on a communication network (internal, external or both depending on how the integrated monitoring system 200 is implemented within the computing system). Each of the monitor modules 202, 204, 206, 208, 210 are connected to the communication network of the computing system as necessary to perform their given function. Examples of monitor modules include firewalls for connection monitoring, dynamic host configuration protocol (DHCP) modules for extracting IP information from the network, intrusion detection systems (IDSs) monitor data traffic and detect attacks, intrusion detection and prevention (IDP) systems that detect and attempt to block attacks, host-based intrusion detection systems (HIDS), proxy and cache servers, forwarders, anti-spam filters, content filters, and virus filters, honey pots, and password protection modules.

The monitor modules monitor the communications traffic to identify messages that may pose a security threat to the computing system. Each monitor module may evaluate the communication traffic in a different way in an attempt to identify different potential threats. Upon identification of a potentially threatening message by an monitor module, the monitor module may take unilateral action to address the threat. In addition to any such unilateral action, the monitor modules also report event data related to the events that are identified.

Each message identified as a potential security threat by one or more of the monitor modules is a single "event." [add alternatives] That is, if a message is identified by several different monitor

modules, possibly for different reasons, as a potential threat, that message will be considered a single event.

Each monitor modules 202, 204, 206, 208, 210 is that they provide data related to the communications traffic on the network. For identified events monitor modules may generate and report data describing or otherwise related to the event. This data, referred to as event data, may be the only indication that the monitor module has identified a potential threat. [possible embodiments}

The event data reported, of course, are dictated by the implementation of the reporting monitor module. Such event data may include, for example, data identifying the monitor module generating the event data, the event type, a priority associated with the event determined by the monitor module, a timestamp for the event, and one or more identifying details of the message that is the source of the event, such as the source IP address, port, URL or MAC of the message, an identifier indicating if the source is internal to the computing system, the destination IP address, port, URL or MAC of the message, an identifier indicating if the destination is internal to the computing system, and information concerning whether the message is coming from a known "bad" or "good" host. The event data may be provided as a simple ASCII file with a known format, as XML that include data type definitions, in an HTML file, or in any other form, as long it is known to and useable by the SSI.

For example, a stateful firewall monitor module that remembers the context of connections and continuously updates this state information in dynamic connection tables, may use one or more IP tables to identify known sources of threats and automatically block traffic from those IP addresses in the IP tables. In the event that messages from IP addresses in the IP tables are identified and blocked by the firewall, the firewall may report event data to the SSI including the source IP address, the destination IP address, identifying information regarding the content of the message, and the date and time the message was received by the firewall.

Another monitor module may be an IDP system. The IDP may include an internal set of rules for use in evaluating and blocking messages in real time. Upon detection of a threat, the IDP system may report an alert, a threat ID and description, a timestamp, and the source and destination IP addresses of the message. Additional event data may also be reported depending on the implementaion.

In the embodiment, the monitor modules 202, 204, 206, 208, 210 report event data to a monitor module integrator (SSI) 212. The event data is received by the SSI 212 and stored in an event database 214. In one embodiment, the SSI 212 maintains the event database 214 so that all event data received from the monitor modules 202, 204, 206, 208, 210 relating to a specific event (i.e. a single message) is collected and stored within a single event record in the event database. In an alternative embodiment, a new event record is created for each item of event data received. In order to prevent the database from getting too big, the event database 214 may purge event data that reach a specified age or may store data until some predetermined database size is reached.

The event database may be structured in various ways. In one embodiment, four different tables are maintained: An Event Priority Table; an Event Type Table; a Log Source Table; and an Event Log Table.

The Event Priority Table includes records related to the event priority. For example, the records include an event priority identifier (EventPriorityID) and a corresponding event priority description associated with that identifier.

The Event Type Table includes records related to the event type. For example, the records include an event type identifier (EventTypeID) and a corresponding event type description associated with the type identifier.

The Log Source Table includes records related to the source of the event. For example, the records include an log source identifier (LogSourceID) and a corresponding event priority description associated with the source identifier.

The final table in the embodiment, the Event Log Table, is the primary repository of the event data. As described above, the event data provided by the monitor modules is stored in event records in the Event Log Table. In addition, various other data generated by the SSI 212 related to the event may also be included in an event record. For example, the SSI may generate unique identifiers for each event record to support future error detection or transmission operations.

TABLE 1, below, includes a list of various event data, along with their descriptions, that may be included in a record, such as an event record, in the tables described above.

TABLE 1 – EVENT DATA

| **Event data type** | **Description** |
|---|---|

| | |
|---|---|
| Event Priority ID | This column is a reference to the primary key for the Event Priority Table. It is used for efficient storage/retrieval and referential integrity checking of valid Event Priority values within the primary (EventLog) table. |
| Event Priority Description | The EventPriority table contains a list of valid Event Priority Descriptions, such as "CRITICAL FIREWALL EVENT". |
| Event Log ID | This is an auto-generated primary key column for efficient storage/retrieval and referential integrity checking of records stored in this table by tables added for to an expanded database. For example, a reporting 'data warehouse' table may refer to records in the EventLog table via this key rather than duplicating data. |
| Log Source ID | In the EventLog table, this column is a reference to the primary key for the LogSource table. It is used for efficient storage/retrieval and referential integrity checking of valid Log Source values within the primary (EventLog) table. |
| Log Source Description | The LogSource table contains a list of valid Log Source Descriptions, such as "FIREWALL". |
| Event Type ID | The type of event, such as a virus contained in an attachment. |
| Event Description | Description of the event, such as for a virus event type the name of the virus identified. |

| | |
|---|---|
| Event Date and Time | A time stamp related to the event, such as when the message was received by the computing system. |
| Source IP | The IP address that the event identifies as its origination point. |
| Source Port | The IP port that a transmission originated from, e.g.: HTTP data generally originates from port 80 |
| Source URL | The uniform resource locator (URL) address that the event identifies as its origination point. |
| Source MAC | This is the Media Access Control address for network devices (a.k.a. nodes). This is a standard unique "ID" for each physical port of network devices such as computer network interface cards, network switching equipment, etc. The Source MAC refers to the ID of the communication packet source device. |
| Internal Source | Data indicating if the origination point of the event is internal to the computing system 200. |
| Destination IP | The IP address that the event identifies as its destination. |
| Destination URL | The uniform resource locator (URL) address that the event identifies as its destination |
| Destination Port | The target IP port for a transmission, e.g.: HTTP data is generally received by port 80. |

| | |
|---|---|
| Destination MAC | This is the Media Access Control address for network devices (a.k.a. nodes). This is a standard unique "ID" for each physical port of network devices such as computer network interface cards, network switching equipment, etc. The Destination MAC refers to the ID of the communication packet recipient device. |
| Internal Destination | Data indicating if the destination point of the event is internal to the computing system 200. |
| Auto Bad Host | Data indicating the corresponding source has been manually entered as a bad host, and should therefore be blocked without further analysis (the "Auto" refers to how the default value of this column is set when not specified). |
| Auto Good Host | Data indicating the corresponding source has been manually entered as a good host, and should therefore be allowed without further analysis (the "Auto" refers to how the default value of this column is set when not specified). |

The SSI 212 includes an analysis module 216. The analysis module 216 analyzes the event data in the event database 214 to identify trends and anomalies in the event data. The analysis module may use various statistical analysis techniques to determine if an event poses a greater threat than that identified by the monitor modules reporting the event data. The analysis module also determines if an event potentially poses a type of threat that the monitor modules are not designed to identify. Upon each receipt of new event data, the analysis module 216 reanalyzes the contents of the event database to determine if the new event data changes the results of its previous analysis.

One example of an analysis performed by the analysis module 216 is a Bayes' Theorem, or Bayesian, analysis. A Bayesian analysis is a statistical procedure that estimates parameters of an

underlying distribution based on an observed distribution. Beginning with a prior distribution, which may be based on anything including an assessment of the relative likelihoods of parameters or the results of non-Bayesian observations, event data is collected and an observed distribution is created. Then a calculation may be made to estimate the likelihood of the observed distribution as a function of parameter values. By multiplying this likelihood function by the prior distribution, a unit probability over all possible values is obtained. This is called the posterior distribution. The mode of the distribution is then the parameter estimate, and probability intervals (the Bayesian analog of confidence intervals) can be calculated using the standard procedure. In embodiments, the Bayesian analysis may be performed on any of the event data provided by monitor modules, such as source IP addresses, to determine a likelihood that messages from a source IP address are threats.

An example of a Bayesian analysis on attacker source IP address event data is as follows. After combining source IP address event data, such as from an IDS, a firewall (possibly from the firewall's IP tables), and a honey pot, in the event database, confidence factors (estimates based on observations) are determined and provided by the monitor modules. These confidence factors can be data either obtained from direct observation of modules, data gathered from 3rd party sources or both. Specifically, these could be Internet global, organization global and/or individual system rates for each confirmed "attacker" profile source IP/Port. Generally, the larger and more diverse datasets will provide better statistical reliability.

There are three estimates ("a priori") needed for the Bayesian Analysis engine:

1) Estimate of the "global" subsystem efficiency in detecting an error (average rate of correct detection for all installations).

2) Estimate of the likelihood that a detected error is an actual error for a given system (True Positive rate).

3) Estimate of the likelihood that a detected error is not an actual error for a given system (False Positive rate).

In one possible embodiment, the estimate of global subsystem efficiency is derived from the overall ("global") body of data collected for these systems if possible (for example, the "Firewall Event" rate of IDS level 1 events reported by all systems using a given firewall for a given chain or

series of chains). The estimate of whether the detected error is an actual error are generated from the True Positive and False Positive observations (for example, the number of level 1 events in IDS log files collected).

For example, given the number of firewall event records for an IP/Port chain that are observed to be actual IDS level 1 events along with the confidence factor we have in IDS accurately reporting true level 1 events, we can use Bayesian Analysis to predict, check upon and make adjustments to the firewall input rules.

*Bayes' Theorem*

$$P(O|D) = P(O) * P(D|O) / (P(O) * P(D|O) + P(O) * P(D|O))$$

Where:

P = Probability

O = Observed Output

D = Observed Data

There are four possible outcomes for a given source IP address tagged as a potential attacker:

1) True Positive (IS an attacker IP and the system reports it as an attacker)

$$P(OT|D=T) = P(OT) * P(D=T|OT) / (P(OT) * P(D=T|OT) + P(OF) * P(D=T|OF))$$

2) False Positive (NOT an attacker but the system reports it as an attacker)

$$P(OF|D=T) = P(OF) * P(D=T|OF) / (P(OT) * P(D=T|OT) + P(OF) * P(D=T|OF))$$

3) False Negative (IS an attacker but the system DOES NOT report it as an attacker)

$$P(OT|D=F) = P(OT) * P(D=F|OT) / (P(OT) * P(D=F|OT) + P(OF) * P(D=F|OF))$$

4) True Negative (NOT an attacker and the system DOES NOT report it as an attacker)

$$P(OF|D=F) = P(OF) * P(D=F|OF) / (P(OT) * P(D=F|OT) + P(OF) * P(D=F|OF))$$

In many environments, IDS is setup to monitor the external interface. As such it is able to view the initiation of possible attacks for all ports, but only use full IDS inspection on ports that are allowed through the firewall (e.g.: ports 22, 25, 80 and 110).

Given this, it may be useful to use Bayesian analysis, for example, to determine from a comparison of IP addresses in IDS level 1 and firewall log records if source IP addresses "attacking" blocked ports can be filtered using an IP specific firewall chain so as to prevent those "attacking" IP's from scanning non-blocked ports.

*Confidence Rate Example*

Rate of "Firewall Event" records matching IDS level 1 class events for ports blocked by firewall input rules: 20% (Output)

Rate of IDS correctly reporting level 1 events on input blocking rules applied to the firewall: 90% (Data)

Given the formulas and confidence rates above would give us:

$P(OT|D=T) = 0.20 * 0.90 / (0.20 * 0.90 + 0.80 * 0.10) = 0.692$ (69.2% True Positives)
$P(OF|D=T) = 0.80 * 0.10 / (0.20 * 0.90 + 0.80 * 0.10) = 0.308$ (30.8% False Positives)

$P(OT|D=F) = 0.20 * 0.10 / (0.20 * 0.10 + 0.80 * 0.90) = 0.027$ ( 2.7% False Negatives)
$P(OF|D=F) = 0.80 * 0.90 / (0.20 * 0.10 + 0.80 * 0.90) = 0.108$ (97.3% True Negatives)

For this example, one could infer from the Bayesian analysis of the firewall and IDS log data that blocking IP addresses from incomplete IDS profiles on ports blocked by firewall would give us about 70.0% confidence of true "hits" (True Positives); however a False Positive rate of about 30% may be unacceptable in an environment where the security device is protecting a heavily used and mission critical public web server installation. The good news in this analysis would be that an inference could be made with a high degree of confidence that if the system doesn't report an event (a Negative), then there isn't one (results in a high True Negative rate).

Note that the "True Positives" confidence factor here would refer to the rate of "hits" against the limiting factor of the 90% efficiency of IDS. In the example above, there would be a 69.2% confidence that a single event would match one of the 90% reported by IDS (.692 * .9 = 62.3% confidence factor against all IDS level 1 events including the 10% not reported).

However, if the IDS efficiency at reporting true level 1 events was raised to 98%, one would could infer the following from Bayesian analysis from this data set:

$$P(OT|D=T) = 0.20 * 0.98 / (0.20 * 0.98 + 0.80 * 0.02) = 0.925 \text{ (92.5\% True Positives)}$$
$$P(OF|D=T) = 0.80 * 0.02 / (0.20 * 0.98 + 0.80 * 0.02) = 0.075 \text{ ( 7.5\% False Positives)}$$

$$P(OT|D=F) = 0.20 * 0.02 / (0.20 * 0.02 + 0.80 * 0.98) = 0.005 \text{ (0.05\% False Negatives)}$$
$$P(OF|D=F) = 0.80 * 0.98 / (0.20 * 0.02 + 0.80 * 0.98) = 0.108 \text{ (99.5\% True Negatives)}$$

It is apparent that as the IDS efficiency closes on 100%, the reliability of the model increases dramatically. Also, a higher rate of "Firewall Event" records that correlate to IDS level 1 events would produce a greater True Positive rate because of the increased probability that any single Firewall Event is a IDS level 1 event. Given a manual override system (e.g.: a "goodhosts" list), a reasonably safe inference could be made from firewall/IDS log data analysis which source addresses to block from all ports and/or protocols.

Additional analyses performed by the analysis module 216 may be designed to identify anomalies and trends in the event records. To do this, the contents of the event database are scanned and events with common data are identified. For example, the analysis will identify event records from common monitoring modules or with common data source/destinations. In addition, the

scanning may also seek to identify known trends indicative of known threats. Events identified with common elements or other known issues are then weighted based on a predetermined weighting algorithm that takes into account the type, priority, monitor module and specifics of the event. The weighting algorithm produces a sum weight for these common events indicating a base severity of the threat (i.e. a threat level). The analysis module 216 then identifies what actions, if any, should be performed based on the calculated threat level. Upon completion of an analysis by the analysis module 216, the results of the analysis may be that the event, and possibly any future messages having specific attributes (for example a point of origination, a destination or specific text in a subject line), should be treated differently by the integrated monitoring system 200 than they are currently being treated. For example, the analysis may determined that every email coming from a certain IP address is likely to be classified as an event by one or more monitor modules and should be screened by the firewall prior to entering the computing system for analysis by the other monitor modules. In these cases, the analysis module 216 may issue commands to other components in the SSI 212. These commands may subsequently be passed, for example by the command and control module 220 as described below, to any connected external component, monitor module or computing system.

In general, the commands allow the SSI 212 to control the operation of any of the other components, modules and devices of the integrated monitoring system 200. The commands issued by the SSI 212 may be as simple as a command to the firewall to add a certain IP address to one or more of its IP tables of IP addresses to block. Other examples of commands include commands to one or more monitor modules that create a new rule to use when evaluating network traffic, commands directing that messages with specific content be allowed to pass, be blocked or be quarantined, commands, such as to a HIDS module, to expand the list of external systems and logs that are evaluated, commands to automatically delete future messages sent to a specified computer port for a specified period of time, and commands changing the threat level assigned by monitor modules to different events. Commands may be issued to the alerting module 218 to generate alerts.

The SSI 212 also includes an alerting module 218. An analysis by the analysis module 216 may determine that a system administrator, various system users, or other designated parties should be alerted to events identified by the SSI 212. In these cases the alerting module 218 identifies the

parties that should be alerted and generates the alert messages with the appropriate data from the event database 214.

The SSI 212 also includes a command and control module 220. The command and control module 220 acts as an interface between the various modules within the SSI 212 and the monitor modules 202, 204, 206, 208, 210. The command and control module 220 stores information concerning how to interface with each monitor module. Using this information, the command and control module can receive a notification, such as from the analysis module 216 for example, that an action by a specific monitor module is required and generate a command for the specific monitor module that carries out the action. Because the command and control module 220 allows the SSI 212 to issue commands to any of the monitor modules capable of receiving commands, an administrator may use the SSI 212 as a central control point for the integrated monitoring system 200.

The SSI 212 is also provided with a communication module 224. The communication module 224 supports the communication between the various other components of the SSI 212 and components and systems external to the SSI 212. In some embodiments, the communication module 224 periodically transmits any new event data received by the event database 214 to a remote computing system or external device for storage or further analysis.

A log database 222 is maintained by the SSI 212 to track actions taken by the SSI 212. The log database 222 may also store log entries recording commands received by the SSI 212 (such as from the administrator) and directed at one or more monitor modules. Other activities may be logged as well depending on the preferences of the system administrator.

FIG. 3 illustrates an embodiment of multi-system implementation of an integrated monitoring system 300. In the embodiment shown, multiple computing systems 302 are interconnected via a communications network 304. Also connected to the communications network 304 is computing system 306 having a security system master integrator (SSMI) 308.

Similar to the operation of the SSI 212, the SSMI 308 receives event data, either directly or indirectly, from a plurality of monitor modules 202, 204, 206, 208. However, the SSMI 308 differs from the SSI 212 in that SSMI 308 receives event data from monitor modules on a plurality of different remote computing systems 302. The event data received then is related to messages that are received at the remote computing systems 302, and not necessarily related to data traffic received by

the computing system 306 having the SSMI 308. In the case of spam messages, for example, the SSMI 308 may receive event data from multiple monitoring modules on computing systems 302 regarding the same spam message, but that message not being one received by the computer system 306 with the SSMI 308. In other respects, such as how the event data is stored and analyzed and how commands are issued, the SSMI 308 operates in a similar fashion as the SSI 212.

The monitor modules of each computing system 302 report event data to the SSMI 308. This event data are generated by the monitor modules at each computing system 302. The event data may be transmitted by the monitor modules directly to the SSMI 308. Alternatively, one or more of the individual computing systems 302 may also include an integrated monitoring system (not shown) such as the one described with reference to FIG. 2 above which may periodically transmit the event data to the SSMI 308. The event data may include the same data as described above in Table 1. In addition, the event data may also include data that identifies the computing system 302 that is transmitting the event data.

The event data are received from each of the computing systems 302 by the SSMI 308 and stored in an event database 314. In embodiments, the event database 314 may include additional event data than is included in the event database 214 in the SSI 212. For instance, the event database 314 may include event data for events that identify the source computing system 302 that generated the event data. In one embodiment, the SSMI 308 maintains the event database 314 so that all event data received from the computing systems 302 relating to a single message are collected and stored within a single event record in the event database. In an alternative embodiment, a new event record is created for each item of event data received. In order to prevent the database from getting too big, the event database 314 may purge event data that reaches a specified age or may store data until some predetermined database size is reached.

The SSMI 308 includes an analysis module 316. The analysis module 316 analyzes the event data in the event database 314 to identify trends and anomalies in the event data in the same manner as the analysis module 216 in the SSI 212. The analysis module uses various statistical analysis techniques to determine if an event poses a greater threat than that identified by the monitor modules reporting the event data. The analysis module also determines if an event potentially poses a type of threat that the monitor modules are not designed to identify. Upon each receipt of an item of event

data, the analysis module 316 reanalyzes the contents of the event database to determine if the new event data changes the results of its previous analysis.

Upon completion of an analysis by the analysis module 316, the results of the analysis may be that the event and any future messages having specific attributes in common with the event (for example a IP address of origination or destination) should be treated differently by one or more of the computing systems 302 than events that are currently being treated. For example, the analysis may determine that every email coming from a certain IP address is infected with a virus. Such information may be particularly important to computing systems 304 that do not have an anti-virus monitor module. In these cases, the analysis module 316 may issue commands to monitor modules in some or all of the computing systems 302.

The commands issued by the SSMI 308 allow the SSMI 308 to control the operation of the monitor modules and other components on the remote computing systems 302. The commands issued may be as simple as a command to one or more firewalls on various computing systems 302 to add a certain IP address to one or more of IP tables. Other examples of commands include commands to one or more monitor modules that create a new rule to use when evaluating network traffic, commands directing that messages with specific content be allowed to pass, be blocked or be quarantined, commands, such as to a HIDS module, to expand the list of external systems and logs that are evaluated, commands to the alerting module 318 to send alerts to one or more computing systems 302 or other destinations, and commands changing the threat level assigned by monitor modules to different events. The SSMI 308 also includes an alerting module 318. An analysis by the analysis module 316 may determine that a system administrator, various system users, or other designated parties should be alerted to event identified by the SSMI 308. In these cases the alerting module 318 identifies the parties that should be alerted and generates the alert messages with the appropriate data from the event database 314.

The SSMI 308 also includes a command and control module 320. The command and control module 320 acts as an interface between the analysis module 316 and the monitor modules in the computing systems 302. The command and control module can receive a notification from the analysis module that an action by a specific monitor module on a specific computer system 302 is required and generate a command for the specific monitor module on that computing system 302 that carries out the action.

Alternatively, if the computing systems 302 include an SSI as described with reference to FIG. 2, the command and control module 320 may issued commands to the SSI which is then responsible for passing the command on the monitor module(s) through its own command and control module.

The SSMI 308 is also provided with a communication module 310. The communication module 310 supports the communication between the various other components of the SSMI 308 and computing systems external to the SSMI 308.

A log database 322 is maintained by the SSMI 308 to track actions taken. Other activities may be logged as well depending on the preferences of the system administrator.

FIG. 4 shows an embodiment of the logical operations of an monitor module on a computing system such as the one described with reference to FIG. 2.

Operational flow begins with a receiving operation 402 in which a new message is received from a communication network being monitored by one or more monitor modules.

In an evaluation operation 404, the message received in the receiving operation 402 is evaluated by one or more monitor modules on the computing system. As discussed above, the nature and extent of the evaluation will vary between the monitor modules. Each monitor module will evaluate the message and either let the message pass or identify the message as potentially posing a threat to the computing system. In addition, an monitor module that identifies the message as a threat may also perform various unilateral actions in response to the evaluation, such as blocking the message's delivery or quarantining the message or a portion of the message.

After the evaluation operation 404, a determination operation 406 identifies if the message was evaluated to be a threat by any of the monitor modules (i.e. if the message qualifies as an "event" or not). If the message is not identified as a threat by any of the monitor modules, the flow terminates in a waiting operation 408 where the monitor module waits for the next message to be received.

If, however, the message is identified as a threat by an monitor module in the determination operation 406, then the message is an "event" and event data is transmitted, in an event data transmitting operation 410, by the monitor module to some destination such as an SSI or an event data log.

In addition to the event data transmitting operation 410, the monitor module may also perform some unilateral action to protect the computing system from the threat posed by the event. This is illustrated by the protective action operation 412.

The event data transmitting operation 410 includes generating event data as well as transmitting the data to some destination such as an SSI. Note that in one embodiment, until event data is transmitted, the destination may be unaware that a threat has been identified. Note also that the message which is the source of the event may already have been deleted or otherwise handled by the monitor module in the protective action operation 412 by the time the event data describing the event is received by the destination.

Upon transmission of the event data and completion of any protective actions, the monitor module method 400 terminates in the waiting operation 408.

FIG. 5 shows an embodiment of the logical operations of an SSI in an integrated monitoring system on a computing system such as the one described with reference to FIG. 2.

Receipt of event data in an event data receiving operation 502 initiates the operation of the SSI. Note that until event data is received, the SSI may be unaware that a threat has been identified by one of the monitor modules. Note also that the message that makes up the event may already have been deleted or otherwise handled by the monitor module by the time the event data describing the event is received by the SSI.

After receipt of the event data, a determination operation 504 determines if there is a record in the event database for the event. This is done by inspecting the event data received to determine if there is an existing event record on the database for the same message. If there already exists an event record for this event, then a data storing operation 508 stores the event data into the event record. If there is not an event record for the event, then a create record operation 506 creates an event record for the event and the data storing operation 508 stores the event data in the event record.

Once the event data is stored in the event record, an analysis operation 510 analyzes the contents of the event database. Depending on the type, priority, monitor module and specifics of the event, the analysis operation scans the database for event records with common data. For example, the analysis will identify event records from common monitoring modules or with common data source/destinations. Once identified, the common event records are then weighted depending on type and priority, with the sum weight for these common events indicating a base severity of the

threat (i.e. a threat level). As the analysis is performed after each receipt of event data from a monitor module, then the analysis will find commonalities between the new event data and the pre-existing event records. In this case, the results of the analysis can be considered specific to the new event data.

Note, however, that in an alternative embodiment, the analysis may be performed periodically or on some other basis, multiple commonalities may be identified for different events received since the last analysis. In this case, different actions may be simultaneously taken in response to different events.

A second determination operation 512 determines, based on the results of the analysis operation 510, if the SSI needs to take action to protect the computing system. In one embodiment, for example, based on the threat level (sum of the weighting), either no action will be taken to interrupt the communication, further statistical analysis will be initiated to better establish a confidence rate for no action, or the communication will be immediately interrupted without further processing. If the second determination operation 512 determines that additional protective actions or measures are warranted, a command operation 514 generates and transmits commands to the monitor modules. Note that the SSI may be aware, through the event data received, of protective actions already performed by the monitor modules. Therefore, protective actions determined by the second determination operation 512 are then actions that are in addition to those already unilaterally performed by the monitor modules. Note also the actions determined may be actions to be performed by one or more monitor modules that heretofore have not identified the event as a threat. For example, the SSI may determine that all messages from a specific IP address should be blocked and may issue a command to a firewall to block such messages even though this action is predicated upon data received from a honey pot or an anti-spam filter.

Command operation 514 may also generate one or more alerts to be transmitted to administrators, logs or other destinations. The specific alert transmitted may be determined by the analysis operation 510 as a function of the type of threat, the threat priority, and other characteristics. The contents of the alert may include various elements of the event data necessary to characterize the event as well as other information such as a description of the protective actions taken unilaterally (if any) by the monitor modules and protective actions taken (if any) by the SSI.

After the action is performed, the SSI goes into a waiting state in a waiting operation 516 in which the SSI waits for new event data. Similarly, if the second determination operation 512 determines that no additional action needs to be performed at this time, then the SSI goes into a waiting state in the waiting operation 516.

The embodiment described in FIG. 5 is but one of a number of possible embodiments that will be suggested to one skilled in the art. For example, in an alternative embodiment analysis operation 510 is performed periodically, rather than after each receipt of event data. In yet another embodiment, the analysis operation 510 is performed only after a delay period that allows for all event data related to a specific event to be received and collected into a single event record. In yet another embodiment, analysis operation 510

FIG. 6 shows an embodiment of the logical operations of an SSMI in a multi-system integrated monitoring system such as the one described with reference to FIG. 3.

The operational flow starts with the receipt of event data in an event data receiving operation 602. Note that until event data is received, the SSMI may be unaware that a threat has been identified by one of the remote computing systems. Note also that the message which is the source of the event may already have been deleted or otherwise handled by the monitor module by the time the event data describing the event is received by the SSMI.

After receipt of the event data, a determination operation 604 determines if there is a record in the event database for the event. This is done by inspecting the event data received to determine if there is an existing event record on the database for the same message. If there already exists an event record for this event, then a data storing operation 608 stores the event data into the event record. If there is not an event record for the event, then a create record operation 606 creates an event record for the event and the data storing operation 608 stores the event data in the event record.

Once the event data is stored in the event record, an analysis operation 510 analyzes the contents of the event database. Depending on the type, priority, monitor module and specifics of the event, the analysis module scans the database for event records with common data. For example, the analysis will identify event records from common monitoring modules or with common data source/destinations. Once identified, the common event records are then weighted depending on type and priority, with the sum weight for these common events indicating a base severity of the threat (i.e. a threat level). As the analysis is performed after each receipt of event data from a

monitor module, then the analysis will find commonalities between the new event data and the pre-existing event records. In this case, the results of the analysis can be considered specific to the new event data.

Note, however, that in an alternative embodiment, the analysis may be performed periodically or on some other basis, multiple commonalities may be identified for different events received since the last analysis. In this case, different actions may be simultaneously taken in response to different events.

A second determination operation 612 determines, based on the results of the analysis operation 610, if the SSMI needs to take action to protect one or more of the remote computing systems. In one embodiment, for example, based on the threat level (sum of the weighting), either no action will be taken to interrupt the communication, further statistical analysis will be initiated to better establish a confidence rate for no action, or the communication will be immediately interrupted without further processing. If the second determination operation 612 determines that additional protective actions or measures are warranted, a command operation 614 generates and transmits commands to the appropriate computing systems. Note that the SSMI may be aware, through the event data received, of protective actions already performed by the monitor module(s) and the SSI (if any) at each of the computing systems. Therefore, protective actions determined by the second determination operation 612 are then actions that are in addition to those already performed by the monitor modules or the SSI at each of the computing systems. Note also the actions that the SSMI determines are to be performed may be actions by one or more monitor modules of remote computing systems that heretofore have not identified the event as a threat, or, indeed, even received such a message. For example, the SSMI may determine that all messages from a specific IP address should be blocked and may issue a command to the firewalls of all the computing systems that have yet to receive messages from the specific IP address.

Command operation 614 may also generate one or more alerts to be transmitted to administrators, logs or other destinations. The specific alert transmitted may be determined by the analysis operation 610 as a function of the type of threat, the threat priority, and other characteristics. The contents of the alert may include various elements of the event data necessary to characterize the event as well as other information such as a description of the protective actions taken unilaterally (if any) by the monitor modules and protective actions taken (if any) by the SSMI.

After the action(s) is performed, the SSMI goes into a waiting state in a waiting operation 616 in which the SSMI waits for new event data. Similarly, if the second determination operation 612 determines that no additional action needs to be performed at this time, then the SSMI goes into a waiting state in the waiting operation 616.

The various embodiments described above are provided by way of illustration only and should not be construed to limit the invention. Those skilled in the art will readily recognize various modifications and changes that may be made to the present invention without following the example embodiments and applications illustrated and described herein, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.